

# Integración de proyectos con réplicas de bases de datos

## Contenido



Integración con “Backup y restore” .....	2
Integración con manejo de transacciones y transacciones anidadas .....	4
Integración con procedimientos y funciones almacenadas .....	6
Integración con vistas indexadas .....	7
Integración con optimización de consultas a través de índices .....	9

## Integración con “Backup y restore”

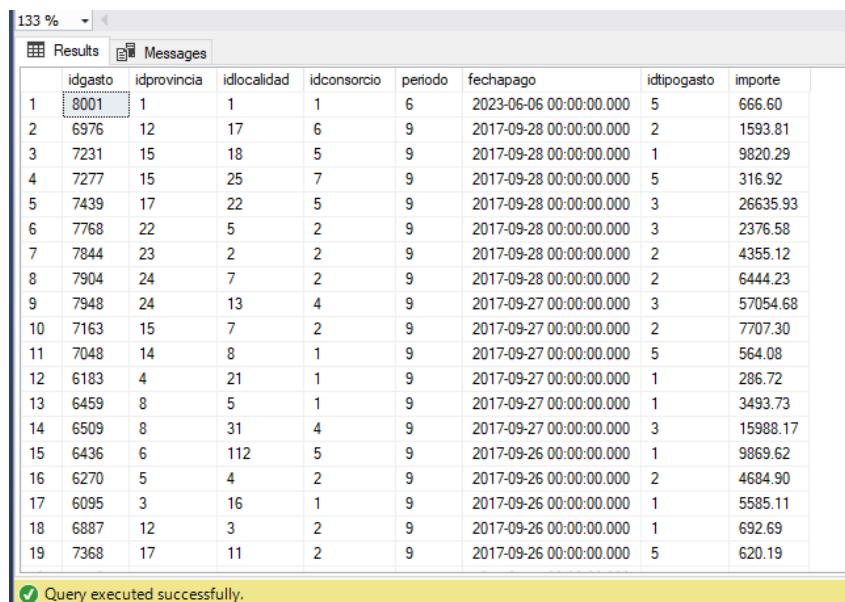
Si bien, el concepto de réplica de bases de datos es similar a hacer un backup en tiempo real en una base de datos alternativa, ésta se hace de forma automática mediante el gestor del servidor de SQL Server.

En esta ocasión, el backup se hace de forma manual y elegimos en cada ocasión en qué lugar de nuestra computadora queremos guardar la copia de seguridad.

Para comenzar con la integración, nos conectamos al servidor donde la base de datos será replicada, luego, insertamos el lote de datos que se nos facilita en el apartado **4.6 Resultados** del proyecto a integrar.

- +  DESKTOP-1V9HDP4 (SQL Server 16.0.1105.1 - PC-Mauricio)
- +  DESKTOP-U7D6G0G (SQL Server 16.0.1000.6 - Mauricio)

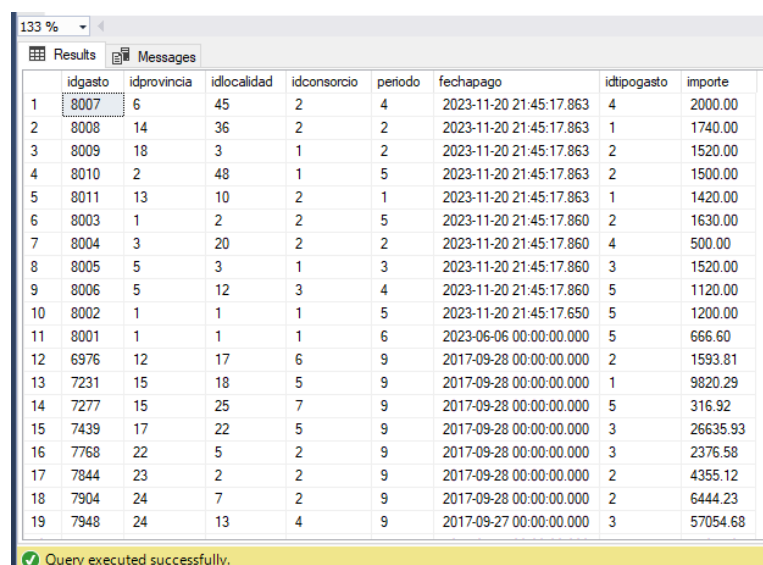
El último gasto agregado en base\_consortio fue en 2023-06-06



	idgasto	idprovincia	idlocalidad	idconsorcio	periodo	fechapago	idtipogasto	importe
1	8001	1	1	1	6	2023-06-06 00:00:00.000	5	666.60
2	6976	12	17	6	9	2017-09-28 00:00:00.000	2	1593.81
3	7231	15	18	5	9	2017-09-28 00:00:00.000	1	9820.29
4	7277	15	25	7	9	2017-09-28 00:00:00.000	5	316.92
5	7439	17	22	5	9	2017-09-28 00:00:00.000	3	26635.93
6	7768	22	5	2	9	2017-09-28 00:00:00.000	3	2376.58
7	7844	23	2	2	9	2017-09-28 00:00:00.000	2	4355.12
8	7904	24	7	2	9	2017-09-28 00:00:00.000	2	6444.23
9	7948	24	13	4	9	2017-09-27 00:00:00.000	3	57054.68
10	7163	15	7	2	9	2017-09-27 00:00:00.000	2	7707.30
11	7048	14	8	1	9	2017-09-27 00:00:00.000	5	564.08
12	6183	4	21	1	9	2017-09-27 00:00:00.000	1	286.72
13	6459	8	5	1	9	2017-09-27 00:00:00.000	1	3493.73
14	6509	8	31	4	9	2017-09-27 00:00:00.000	3	15988.17
15	6436	6	112	5	9	2017-09-26 00:00:00.000	1	9869.62
16	6270	5	4	2	9	2017-09-26 00:00:00.000	2	4684.90
17	6095	3	16	1	9	2017-09-26 00:00:00.000	1	5585.11
18	6887	12	3	2	9	2017-09-26 00:00:00.000	1	692.69
19	7368	17	11	2	9	2017-09-26 00:00:00.000	5	620.19

Query executed successfully.

Insertamos el lote de datos en nuestra base de datos principal, y vemos los datos actualizados.



	idgasto	idprovincia	idlocalidad	idconsorcio	periodo	fechapago	idtipogasto	importe
1	8007	6	45	2	4	2023-11-20 21:45:17.863	4	2000.00
2	8008	14	36	2	2	2023-11-20 21:45:17.863	1	1740.00
3	8009	18	3	1	2	2023-11-20 21:45:17.863	2	1520.00
4	8010	2	48	1	5	2023-11-20 21:45:17.863	2	1500.00
5	8011	13	10	2	1	2023-11-20 21:45:17.863	1	1420.00
6	8003	1	2	2	5	2023-11-20 21:45:17.860	2	1630.00
7	8004	3	20	2	2	2023-11-20 21:45:17.860	4	500.00
8	8005	5	3	1	3	2023-11-20 21:45:17.860	3	1520.00
9	8006	5	12	3	4	2023-11-20 21:45:17.860	5	1120.00
10	8002	1	1	1	5	2023-11-20 21:45:17.650	5	1200.00
11	8001	1	1	1	6	2023-06-06 00:00:00.000	5	666.60
12	6976	12	17	6	9	2017-09-28 00:00:00.000	2	1593.81
13	7231	15	18	5	9	2017-09-28 00:00:00.000	1	9820.29
14	7277	15	25	7	9	2017-09-28 00:00:00.000	5	316.92
15	7439	17	22	5	9	2017-09-28 00:00:00.000	3	26635.93
16	7768	22	5	2	9	2017-09-28 00:00:00.000	3	2376.58
17	7844	23	2	2	9	2017-09-28 00:00:00.000	2	4355.12
18	7904	24	7	2	9	2017-09-28 00:00:00.000	2	6444.23
19	7948	24	13	4	9	2017-09-27 00:00:00.000	3	57054.68

Query executed successfully.

Vemos que en la réplica de la base de datos se hicieron los cambios correctamente.

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The Object Explorer on the left displays the server hierarchy for 'DESKTOP-U7D6G0G (SQL Server 16.0.1000.6 - Mauricio)'. The 'rep\_base\_consorcio' database is selected. The SQL Query window on the right shows the following query:

```
SELECT * FROM GASTO
ORDER BY fechapago DESC;
```

The Results pane at the bottom displays the query output as a table with 10 rows. The status bar indicates 'Query executed successfully' and '8,010 rows'.

	idgasto	idprovincia	idlocalidad	idconsorcio	periodo	fechapago	idtipogasto	importe
1	8007	6	45	2	4	2023-11-20 21:45:17.863	4	2000.0
2	8008	14	36	2	2	2023-11-20 21:45:17.863	1	1740.0
3	8009	18	3	1	2	2023-11-20 21:45:17.863	2	1520.0
4	8010	2	48	1	5	2023-11-20 21:45:17.863	2	1500.0
5	8011	13	10	2	1	2023-11-20 21:45:17.863	1	1420.0
6	8003	1	2	2	5	2023-11-20 21:45:17.860	2	1630.0
7	8004	3	20	2	2	2023-11-20 21:45:17.860	4	500.00
8	8005	5	3	1	3	2023-11-20 21:45:17.860	3	1520.0
9	8006	5	12	3	4	2023-11-20 21:45:17.860	5	1120.0
10	8002	1	1	1	5	2023-11-20 21:45:17.650	5	1200.0

Ahora procedemos a realizar el backup en la máquina virtual de la réplica de la base de datos.

La ruta será: C:\Users\Contr\OneDrive\Documents\ContralXosh\Virtual Machine Backup

## Integración con manejo de transacciones y transacciones anidadas

Para este trabajo, utilizaremos las transacciones facilitadas por los compañeros en los casos que las transacciones son exitosas y pueden terminar en su totalidad.

Se insertarán:

- Un valor en la tabla administrador
- Un valor en la tabla consorcio
- Un valor en la tabla gasto

Verificamos, en nuestra base de datos principal, los últimos datos ingresados:

- En la tabla administrador, el último administrador agregado fue “Mauricio Emmanuel Villalba”

```
SELECT * FROM administrador
ORDER BY idadmin DESC;
```

	idadmin	apeynom	viveahi	tel	sexo	fechnac
1	175	Mauricio Emmanuel Villalba	S	3794999999	M	2001-03-16 00:00:00.000
2	174	ARAUJO GUILLERMO JOSE	S	3672235689	M	1985-10-13 00:00:00.000

- En la tabla consorcio, observamos que no existe un consorcio con dirección ‘PARAGUAY N 999’

```
SELECT * FROM consorcio
WHERE direccion = 'PARAGUAY N 999'
```

idprovincia	idlocalidad	idconsorcio	nombre	direccion	idzona	idconserje	idadmin	idtipopropiedad
-------------	-------------	-------------	--------	-----------	--------	------------	---------	-----------------

- En la tabla gasto, vemos que existe un solo gasto de importe 608.97

```
SELECT * FROM gasto
WHERE importe = 608.97
```

idgasto	idprovincia	idlocalidad	idconsorcio	periodo	fechapago	idtipogasto	importe
1	1	1	1	6	2013-06-16 00:00:00.000	5	608.97

Ejecutamos la siguiente transacción en nuestra base de datos principal:

```
--- CASO Transacción Anidada
-- USE base_consortio;

-- CREACION DE TRANSACCIONES
BEGIN TRY -- INICIAMOS EL BEGIN TRY PARA LUEGO COLOCAR LA LOGICA DENTRO Y ASEGURARNOS DE QUE SI ALGO FALLA IRA POR EL CATCH
BEGIN TRAN -- COMENZAMOS LA TRANSACCION
INSERT INTO administrador(apeynom, viveahi, tel, sexo, fechnac) -- UN INSERT A LA TABLA QUE QUEREMOS
VALUES ('pablito clavounclavito', 'S', '37942222', 'M', '01/01/1996') -- LOS VALORES A INGRESAR A LA TABLA

INSERT INTO consorcio(idprovincia, idlocalidad, idconsorcio, nombre, direccion, idzona, idconserje, idadmin)
VALUES (1, 1, 3, 'EDIFICIO-111', 'PARAGUAY N 999', 5, 100, 1) -- GENERAMOS UN ERROR INGRESANDO EL ID PROVINCIA 999 QUE NO EXISTE.

-----
BEGIN TRAN -- COMENZAMOS UNA TRANSACCION ANIDADA
UPDATE consorcio SET nombre = 'EDIFICIO-222'; -- ACTUALIZAMOS EL REGISTRO DE CONSORCIO QUE CARGAMOS ANTES
INSERT INTO gasto(idprovincia, idlocalidad, idconsorcio, periodo, fechapago, idtipogasto, importe)
VALUES (1,1,1,6,'20130616',5,608.97) -- INSERT A LA TABLA GASTO, DEBE DAR UN ERROR POR TIPO DE GASTO
COMMIT TRAN -- FINALIZAMOS UNA TRANSACCION ANIDADA
-----

INSERT INTO gasto(idprovincia, idlocalidad, idconsorcio, periodo, fechapago, idtipogasto, importe)
VALUES (1,1,1,6,'20130616',5,608.97)
INSERT INTO gasto(idprovincia, idlocalidad, idconsorcio, periodo, fechapago, idtipogasto, importe)
VALUES (1,1,1,6,'20130616',2,608.97)
INSERT INTO gasto(idprovincia, idlocalidad, idconsorcio, periodo, fechapago, idtipogasto, importe)
VALUES (1,1,1,6,'20130616',3,608.97)

COMMIT TRAN -- SI TODO FUE EXITOSO FINALIZAMOS LA TRANSACCION
END TRY
BEGIN CATCH -- SI ALGO FALLA VENDRA AQUI
SELECT ERROR_MESSAGE() -- MOSTRAMOS EL MENSAJE DE ERROR
ROLLBACK TRAN -- VOLVEMOS HACIA ATRAS PARA MANTENER LA CONSISTENCIA DE LOS DATOS
END CATCH
```

La transacción se realiza correctamente, y observamos en nuestra base de datos replicada que la transacción se finalizó por completo.

Object Explorer: Connect to DESKTOP-U7D6G0G (SQL Server 16.0.1105.1 - sa)

SQL Query1.sql - DES...consorcio (sa (58))

```
-- NOS MUESTRA CUAL FUE EL ULTIMO REGISTRO DE ADMINISTRADOR CARGADO
SELECT TOP 1 * FROM administrador ORDER BY idadmin DESC;

-- MUESTRA LOS DATOS DEL CONSORCIO CON LA DIRECCION EN CUESTION (EXITE O NO)
SELECT * FROM consorcio WHERE direccion = 'PARAGUAY N 999';

-- MUESTAR LOS ULTIMOS 3 REGISTROS DE GASTOS CARGADOS
SELECT TOP 4 * FROM gasto ORDER BY idgasto DESC;
```

Results

idadmin	apeynom	viveahi	tel	sexo	fechnac
1178	pablito clavounclavito	S	37942222	M	1996-01-01 00:00:00.000

idprovincia	idlocalidad	idconsorcio	nombre	direccion	idzona	idconserje	idadmin
1	1	4	EDIFICIO-222	PARAGUAY N 999	5	100	1
2	1	5	EDIFICIO-222	PARAGUAY N 999	5	100	1

idgasto	idprovincia	idlocalidad	idconsorcio	periodo	fechapago	idtipogasto	importe
8018	1	1	1	6	2013-06-16 00:00:00.000	3	608.97
8017	1	1	1	6	2013-06-16 00:00:00.000	2	608.97
8016	1	1	1	6	2013-06-16 00:00:00.000	5	608.97
8015	1	1	1	6	2013-06-16 00:00:00.000	5	608.97

Query executed successfully | DESKTOP-U7D6G0G (16.0 RTM) | sa (58) | rep\_base\_consortio | 00:00:00 | 175 rows

## Integración con procedimientos y funciones almacenadas

En primer lugar, creamos la función almacenada facilitada por los compañeros. Probaremos con la función para agregar un consorcio a la tabla de consorcios.

Ejecutamos el siguiente script para crear la función almacenada:

```
USE base_consortio
GO

CREATE PROCEDURE agregarConsortio
@Provincia INT,
@Localidad INT,
@Consortio INT,
@Nombre VARCHAR(50),
@Direccion VARCHAR(255),
@Zona INT,
@Conserje INT,
@Admin INT
AS
BEGIN
INSERT INTO consorcio(idprovincia,idlocalidad,idconsorcio,nombre,direccion, idzona,idconserje,idadmin)
VALUES (@Provincia,@Localidad,@Consortio,@Nombre,@Direccion, @Zona, @Conserje, @Admin)
END
```

Como podemos ver, en la base de datos principal, no existe ningún consorcio con nombre Nicolás:

```
SELECT * FROM consorcio
where nombre = 'Nicolás'
```

Results								
idprovincia	idlocalidad	idconsorcio	nombre	direccion	idzona	idconserje	idadmin	idtipopropiedad

Ahora se procede a ejecutar la siguiente llamada a la función para agregar un consorcio:

```
EXECUTE agregarConsortio @Provincia=5, @Localidad=6, @Consortio= 50, @Nombre='Nicolás', @Direccion='General Viamonte 1658', @Zona=3,@Conserje='2',@Admin=1
SELECT * FROM consorcio WHERE idConsortio=50
```

Observamos que la ejecución del procedimiento almacenado realizado en la base de datos principal, se reflejó correctamente en la base de datos replicada.

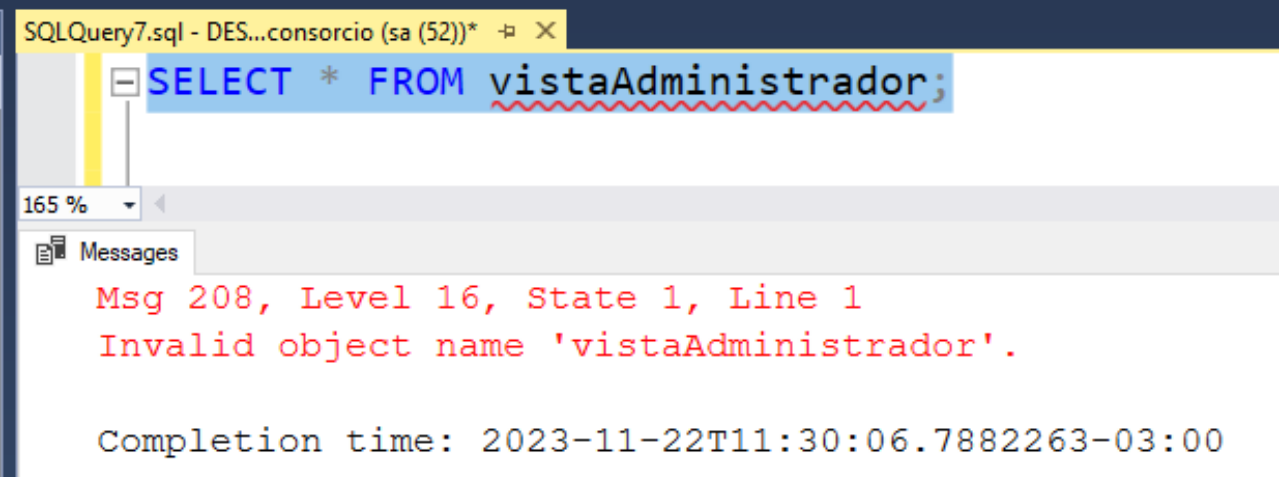
The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the server structure for 'DESKTOP-UTD660G (SQL Server 16.0.1105.1 - sa)'. The 'Databases' folder is expanded, showing 'rep\_base\_consortio'. The 'Messages' pane at the bottom shows the execution of the 'agregarConsortio' stored procedure, indicating it was successful. The 'Results' pane displays the output of the query 'SELECT \* FROM consorcio WHERE idConsortio=50', showing a single row with the following values:

idprovincia	idlocalidad	idconsorcio	nombre	direccion	idzona	idconserje	idadmin	idtipopropiedad
5	6	50	Nicolás	General Viamonte 1658	3	2	1	NU

## Integración con vistas indexadas

Para la integración de este proyecto, y para poder apreciar la réplica transaccional en acción, realizaremos la inserción de nuevos datos, registros y vistas en la base de datos principal para que sean replicadas, y las consultas de las vistas las realizaremos en la réplica de la base de datos.

Primero, en nuestra base de datos principal, verificamos que no existe ninguna vista llamada “vistaAdministrador”



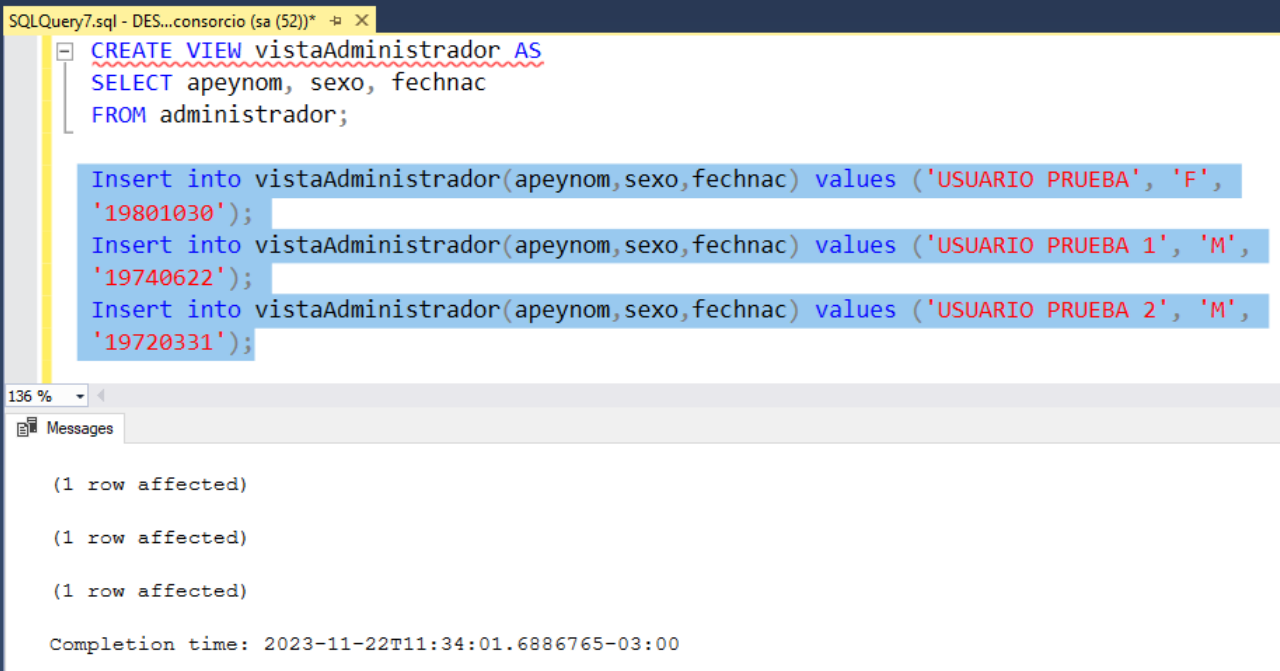
The screenshot shows a SQL query window titled "SQLQuery7.sql - DES...consorcio (sa (52))\*". The query is `SELECT * FROM vistaAdministrador;`. Below the query, the "Messages" pane displays the following error: "Msg 208, Level 16, State 1, Line 1 Invalid object name 'vistaAdministrador'." The completion time is "2023-11-22T11:30:06.7882263-03:00".

```
SQLQuery7.sql - DES...consorcio (sa (52))* X
SELECT * FROM vistaAdministrador;

165 %
Messages
Msg 208, Level 16, State 1, Line 1
Invalid object name 'vistaAdministrador'.

Completion time: 2023-11-22T11:30:06.7882263-03:00
```

En la base de datos principal, creamos la vista “vistaAdministrador” e insertamos el lote de datos para la nueva vista facilitada por los compañeros.



The screenshot shows a SQL query window titled "SQLQuery7.sql - DES...consorcio (sa (52))\*". The query is `CREATE VIEW vistaAdministrador AS SELECT apeynom, sexo, fechnac FROM administrador;` followed by three `INSERT` statements. Below the query, the "Messages" pane displays the following messages: "(1 row affected)", "(1 row affected)", and "(1 row affected)". The completion time is "2023-11-22T11:34:01.6886765-03:00".

```
SQLQuery7.sql - DES...consorcio (sa (52))* X
CREATE VIEW vistaAdministrador AS
SELECT apeynom, sexo, fechnac
FROM administrador;

Insert into vistaAdministrador(apeynom,sexo,fechnac) values ('USUARIO PRUEBA', 'F',
'19801030');
Insert into vistaAdministrador(apeynom,sexo,fechnac) values ('USUARIO PRUEBA 1', 'M',
'19740622');
Insert into vistaAdministrador(apeynom,sexo,fechnac) values ('USUARIO PRUEBA 2', 'M',
'19720331');

136 %
Messages
(1 row affected)
(1 row affected)
(1 row affected)

Completion time: 2023-11-22T11:34:01.6886765-03:00
```

Finalizada la carga en la base de datos principal, nos dirigimos a la réplica de la base de datos para realizar la consulta sobre la nueva vista.

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'rep\_base\_consortorio' database selected. The right pane shows a query window with the following SQL query:

```
SELECT * FROM vistaAdministrador;
```

The query results are displayed in a table with the following columns: **apexnom**, **sexo**, and **fechnac**. The results list 17 rows of data, starting with 'Perez Juan Manuel' and ending with 'VILLALBA SILVIA M'.

	apexnom	sexo	fechnac
1	Perez Juan Manuel	M	1985-02-18 00:00:00.000
2	BASUALDO DELMIRA	F	1980-10-09 00:00:00.000
3	SEGOVIA ALEJANDRO H.	M	1974-06-02 00:00:00.000
4	ROMERO ELEUTERIO	M	1972-08-19 00:00:00.000
5	NAHMIA DE K. NIDIA	F	1971-11-28 00:00:00.000
6	CORREA DE M. MARIA G.	F	1990-01-16 00:00:00.000
7	NAHMIA JOSE	M	1974-09-02 00:00:00.000
8	NAHMIA DE R. REBECA J.	F	1989-03-07 00:00:00.000
9	LOVATO CERENTINI ISABEL	F	1973-10-15 00:00:00.000
10	GOMEZ MATIAS GABRIEL	M	1974-03-20 00:00:00.000
11	CORREA HUGO E.	M	1993-08-11 00:00:00.000
12	MACHUCA CEFERINA	F	1991-09-16 00:00:00.000
13	CARDOZO MAXIMA	F	1988-11-07 00:00:00.000
14	RODRIGUEZ MARTIN J.	M	1985-12-09 00:00:00.000
15	SOTELO GERTRUDIS	F	1990-11-12 00:00:00.000
16	AYALA FLORENTINA	F	1974-04-27 00:00:00.000
17	VILLALBA SILVIA M	M	1994-04-01 00:00:00.000

The status bar at the bottom indicates: 'Query executed suc... | DESKTOP-U7D6G0G (16.0 RTM) | sa (51) | rep\_base\_consortorio | 00:00:00 | 175 rows'.



## Integración con optimización de consultas a través de índices

Usamos el script facilitado por los compañeros en la base de datos replicada integrar el tema.

-- Creación de índices tradicionales para optimización de búsquedas

```
CREATE INDEX idx_nombre_consortio ON consorcio(nombre);
```

-- Creación de índices columnares para mejorar la eficiencia de ciertas consultas

```
CREATE INDEX idx_columnares_inmueble ON inmueble(idprovincia, idlocalidad, idconsorcio, nro_piso);
```

-- Creación de roles y asignación de privilegios para el manejo de usuarios

```
CREATE ROLE db_manager;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON consorcio TO db_manager;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON inmueble TO db_manager;
```

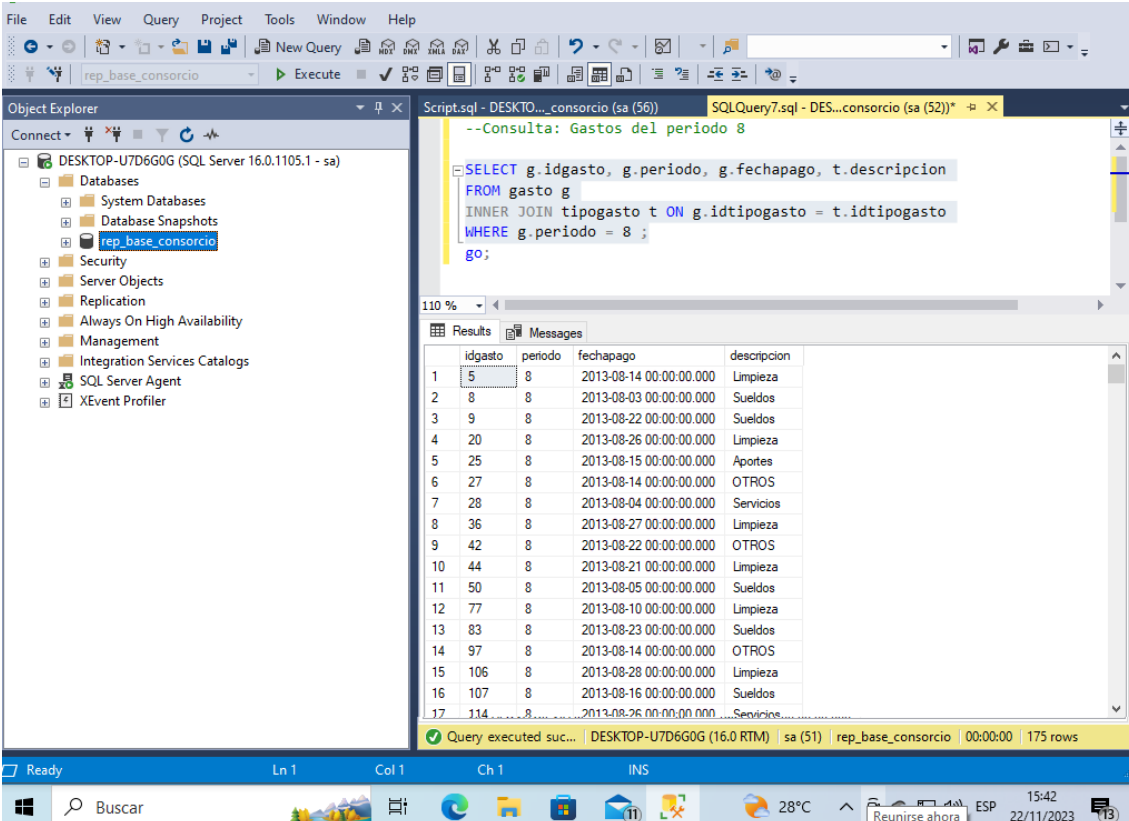
-- Creación de un usuario con acceso limitado y optimización de consultas SQL

```
CREATE USER 'usuario_consulta' WITH PASSWORD 'clave_segura';
```

```
GRANT SELECT ON consorcio TO 'usuario_consulta';
```

```
GRANT SELECT ON inmueble TO 'usuario_consulta';
```

Luego de ejecutar el script en la base de datos replicada, realizamos una consulta y obtuvimos el siguiente resultado en la base de datos replicada:



The screenshot displays the SQL Server Enterprise Manager interface. The Object Explorer on the left shows the database structure, including the 'rep\_base\_consortio' database. The central pane shows a SQL query being executed in the 'rep\_base\_consortio' database. The query is as follows:

```
--Consulta: Gastos del periodo 8
SELECT g.idgasto, g.periodo, g.fechapago, t.descripcion
FROM gasto g
INNER JOIN tipogasto t ON g.idtipogasto = t.idtipogasto
WHERE g.periodo = 8 ;
go;
```

The Results pane shows the output of the query, which consists of 17 rows. The columns are 'idgasto', 'periodo', 'fechapago', and 'descripcion'. The data is as follows:

idgasto	periodo	fechapago	descripcion
5	8	2013-08-14 00:00:00.000	Limpieza
8	8	2013-08-03 00:00:00.000	Sueldos
9	8	2013-08-22 00:00:00.000	Sueldos
20	8	2013-08-26 00:00:00.000	Limpieza
25	8	2013-08-15 00:00:00.000	Aportes
27	8	2013-08-14 00:00:00.000	OTROS
28	8	2013-08-04 00:00:00.000	Servicios
36	8	2013-08-27 00:00:00.000	Limpieza
42	8	2013-08-22 00:00:00.000	OTROS
44	8	2013-08-21 00:00:00.000	Limpieza
50	8	2013-08-05 00:00:00.000	Sueldos
77	8	2013-08-10 00:00:00.000	Limpieza
83	8	2013-08-23 00:00:00.000	Sueldos
97	8	2013-08-14 00:00:00.000	OTROS
106	8	2013-08-28 00:00:00.000	Limpieza
107	8	2013-08-16 00:00:00.000	Sueldos
114	8	2013-08-26 00:00:00.000	Servicios

The status bar at the bottom indicates that the query was executed successfully, returning 175 rows.